

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently Amended) An inter-processor communication apparatus of a mobile communication system comprising:
  - a data-FIFO configured to be directly coupled to a transmission bus and configured to store a receiving data from a master;
  - a slave-logic configured to control a writing operation of the receiving data for the data-FIFO and count ~~[[the]]~~ a length of the receiving data until an end-tag signal is inputted;
  - a length-FIFO directly connected to the slave-logic and configured to store the data length counted by the slave-logic; and
  - a CPU separately connected to the slave-logic, the data-FIFO and the length-FIFO and configured to continuously read the data stored in the data-FIFO as much as the data read from the length-FIFO when an interrupt signal is inputted from the slave-logic.
2. (Original) The apparatus of claim 1, wherein the slave-logic counts the length of the receiving data until an end tag signal is inputted.

3. (Original) The apparatus of claim 2, wherein the read data length is one frame data length.
4. (Original) The apparatus of claim 1, wherein the slave-logic stores the counted data length in the length-FIFO when the end tag signal is inputted and outputs an interrupt signal to the CPU.
5. (Original) The apparatus of claim 1, wherein the CPU continuously reads the data stored in the data-FIFO by 1 byte unit as much as the data length stored in the length-FIFO.
6. (Currently Amended) An inter-processor communication method of a mobile communication system, comprising ~~the steps of~~:
  - storing a ~~receiving~~ data received from a master in a first region;
  - counting ~~[[the]]~~ a length of the receiving data stored in the first region;
  - checking whether an end tag is received;
  - storing the counted data length in a second region when the end tag is receivedand outputting an interrupt signal to a CPU; and

continuously reading the data stored in the first region by the CPU as much as the data length stored in the second region,

wherein a slave-logic is connected to the first region and the second region and configured to control said storing the receiving data and said counting the length of the receiving operation until the end-tag is received, and the CPU is separately connected to the slave-logic, the first region and the second region, and

wherein when the master transmits a data, an end tag and a write signal to the slave-logic:

the slave-logic outputs a write signal to the first region to store the receiving data and the end tag in the first region;

the slave-logic counts the receiving data, and after receiving the end tag, the slave-logic outputs the interrupt signal to the CPU and outputs the counted data length together with a write signal to the second region; and

the CPU reads the data length from the second region based on the interrupt signal and continuously reads the data stored in the first region up to the end-tag.

7. (Original) The method of claim 6, wherein the first and the second regions are FIFO.

8. (Original) The method of claim 6, wherein the data length stored in the second region is one frame of data length.

9. (Original) The method of claim 6, wherein the CPU continuously reads the data by 1 byte unit.

10. (Currently Amended) An inter-processor communication apparatus of a mobile communication system, comprising:

means for storing ~~received~~ data received from a master in a first region;

means for counting a length of the received data stored in the first region and for checking whether an end tag of the received data is received, wherein the means for counting counts until the end tag is received;

means for storing the counted data length in a second region when the end tag is received and outputting an interrupt signal to a CPU; and

~~means~~ the CPU for reading the data stored in the first region ~~by the CPU~~ as much as the data length stored in the second region,

wherein the CPU is separately connected to the means for counting, the means for storing the received data and the means for storing the counted data length, and

wherein the means for counting is directly connected to the first region and the second region.

11-13. (Canceled)

14. (Previously Presented) The apparatus of claim 1, wherein the counted data length of data is read in a single operation from the first region.

15. (Previously Presented) The method of claim 6, wherein the continuously reading comprises performing a signal read operation to read the counted data length of data from the first region.

16. (New) The apparatus of claim 1, wherein when the master transmits a data, an end-tap signal and a write signal to the slave-logic:

the slave-logic outputs a write signal to the data-FIFO to store the receiving data and the end-tap signal in the data-FIFO;

the slave-logic counts the receiving data, and after receiving the end-tap signal, the slave-logic outputs an interrupt signal to the CPU and outputs the counted data length together with a write signal to the length-FIFO; and

the CPU reads the data length from the length-FIFO based on the interrupt and continuously reads the data stored in the data-FIFO up to the end-tag signal.

17. (New) The apparatus of claim 10, wherein when the master transmits a data, an end tag and a write signal to the means for counting:

the means for counting outputs a write signal to the means for storing the received data to store the receiving data and the end tag in the means for storing the received data;

the means for counting counts the receiving data, and after receiving the end tag, the means for counting outputs an interrupt signal to the CPU and outputs the counted data length together with a write signal to the means for storing the counted data length; and

the CPU reads the data length from the means for storing the counted data length based on the interrupt signal and continuously reads the data stored in the means for storing the received data up to the end tag.